

Gustavo Guijarro

Ramy Kassam

Zach Remer

OOAD CSCI 4448 - Project 7

## **Final Report**

**Project Name:** Dough Masters

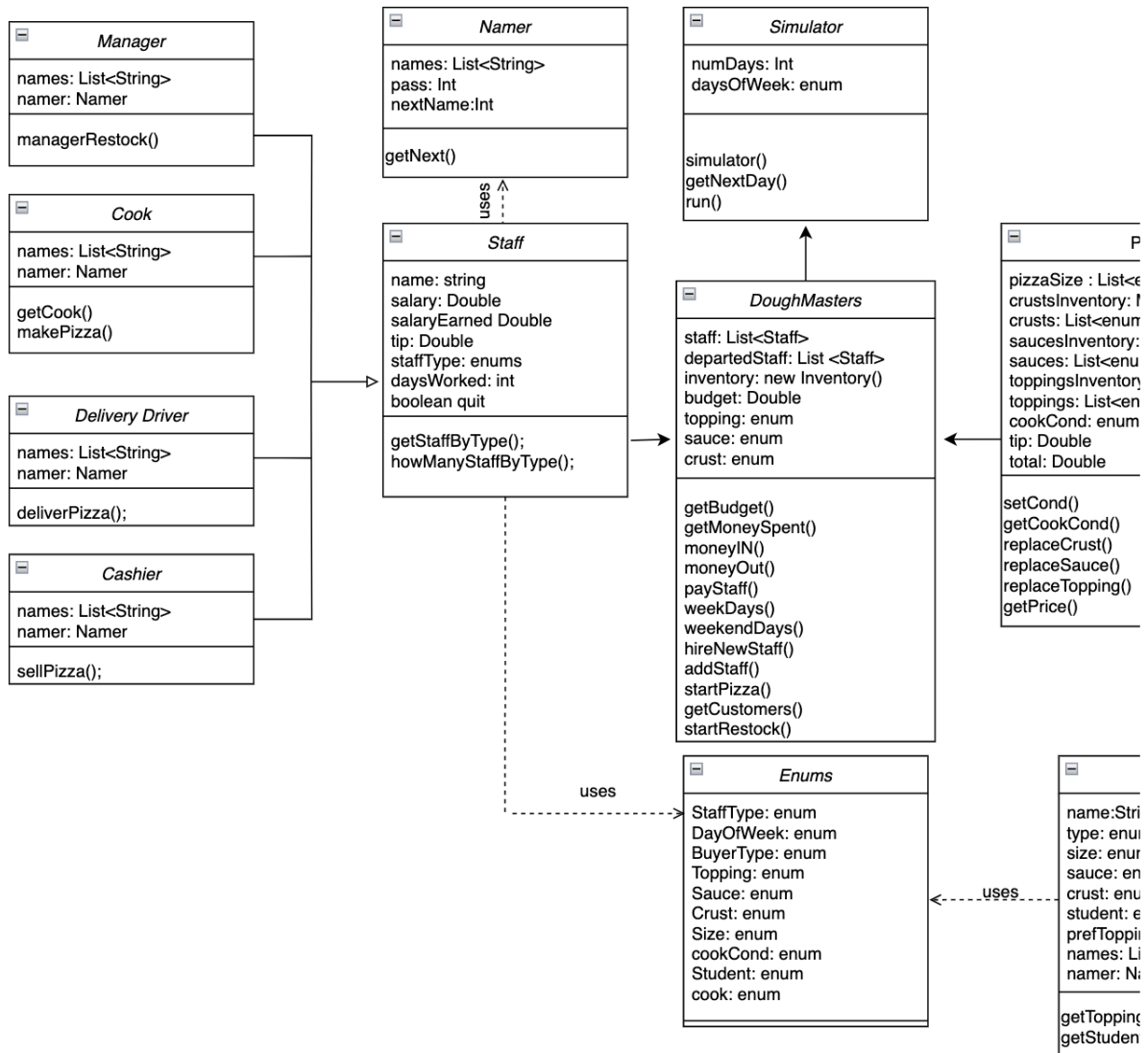
### **Final State of System:**

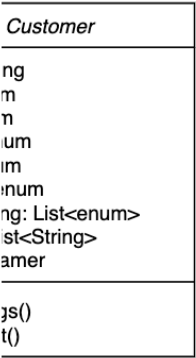
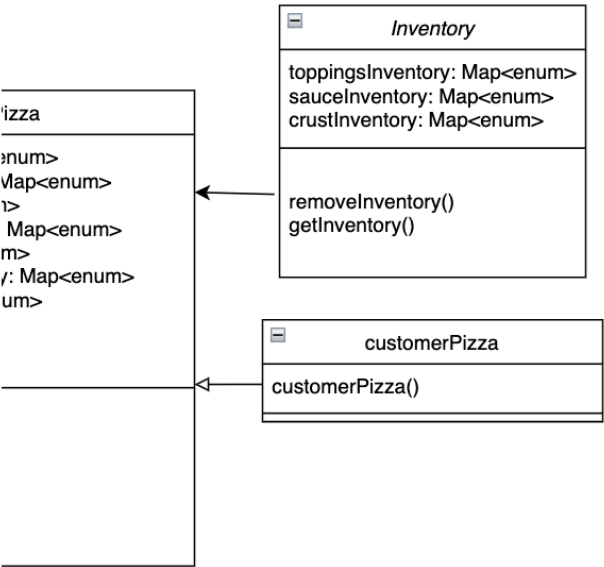
Since projects 5 and 6, much has been implemented in our final system. Previously, we were under-prepared and only implemented one design pattern, the factory pattern. Since then we have used three more design patterns which include strategy pattern, decorator pattern, and observer pattern. We utilized the strategy pattern to initialize the different types of cooks in the pizzeria which were made up of bad cooks, average cooks, and expert cooks. All of which had different traits and usage when making the pizza. For example, a bad cook has a higher chance of undercooking or overcooking a pizza, and the total price of the pizza will be updated accordingly, to compensate the customer. The overall condition of a pizza (how well it is made) influenced the tip received. We also decided to implement add-ons to the pizza order, for instance, if a customer wanted extra meat toppings, extra cheese, extra sauces, or wanted a fast order where their pizza order will take priority and be made faster. All add-ons have a different chance of being chosen and would update/increase the total price of the pizza order. We implemented these add-ons using the decorator pattern. [Observer pattern]. Furthermore, we chose to have pizza discounts on Fridays for students. On these Fridays, students would receive a 10% discount on their entire order.

A Figma prototype modeling the ordering system for our pizza shop has also been completed. Previously, only 2 pages were developed, the login and menu pages. Since Project 6, we have not only polished the login and menu pages but also implemented an ordering page for both pickup and delivery orders and implemented the extra sides customers can add (to model our code more accurately). Furthermore, we made our prototype more 'aesthetically pleasing' to make it look more professional. In summary, the Figma prototype models a phone application where you can order pizzas from our pizzeria shop, Dough Masters.

## Final Class Diagram and Comparison Statement:

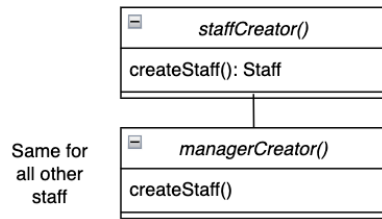
### Final Class Diagram: (Split into two halves for fitting)



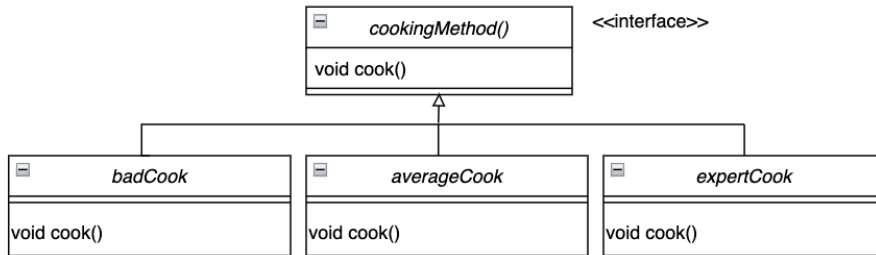


## Pattern Use:

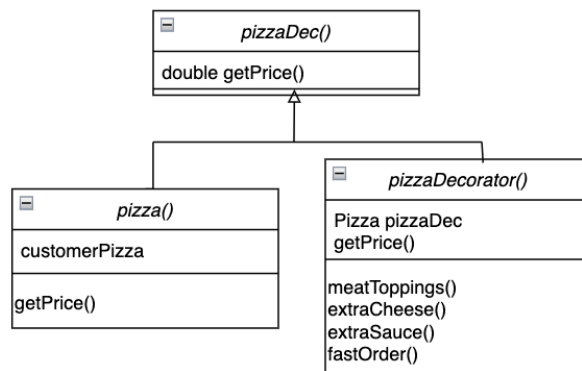
### (simple) Factory Pattern for Staff & Pizzas <<interface>>



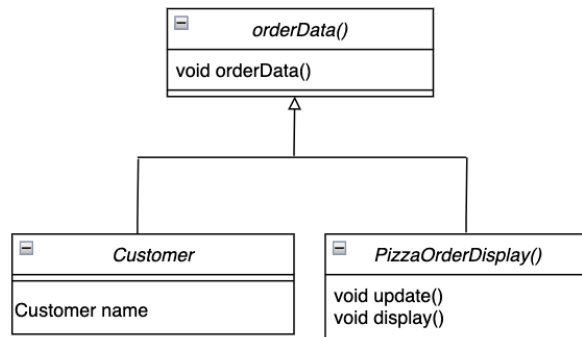
### Strategy Pattern



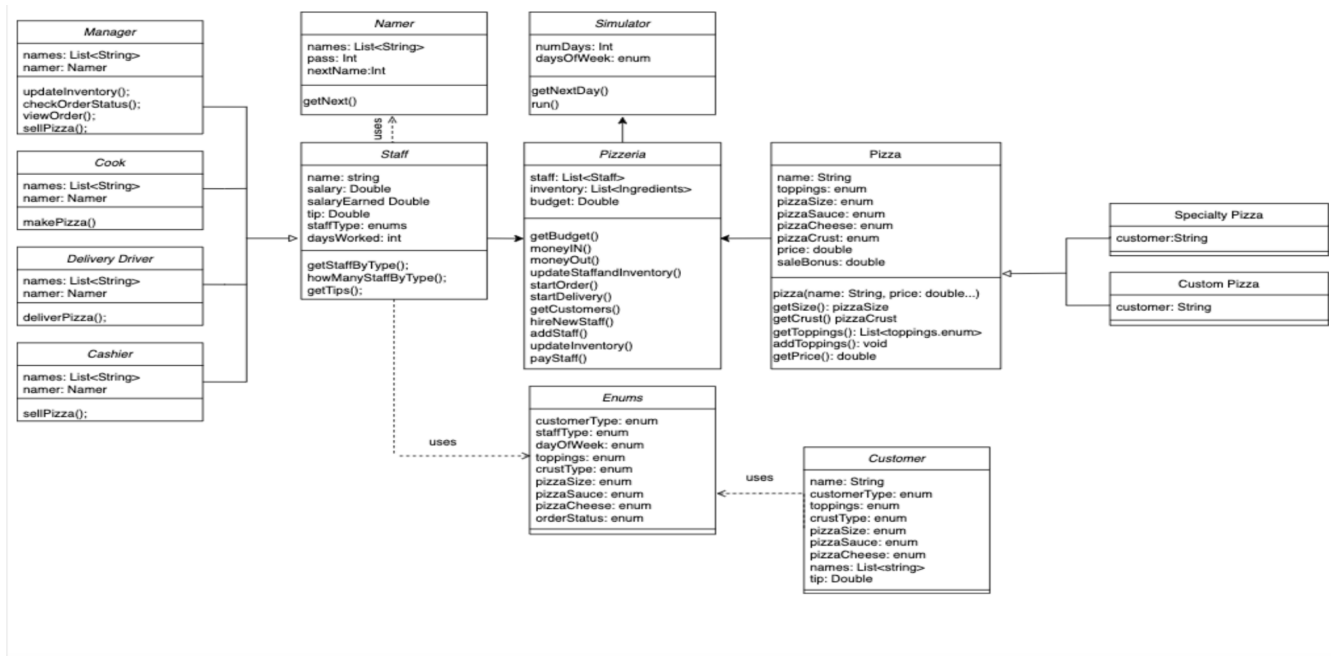
### Decorator Pattern



### Observer Pattern



## Project 5 Class Diagram:



Since projects 5 and 6, there have been many changes/additions to our system. These changes are most obvious when comparing our class diagram from Project 5 to our class diagram right now. For instance, at a quick glance, we can visualize how the length of our previous classes has gotten noticeably longer. Project 5 was more of a planning phase for our system so we did not hash out all the specific details. When we began to implement our system we realized we needed more functions and classes. Some key changes included making an entirely new class for managing the inventory for ingredients or when we started to use the strategy pattern for making pizzas, we had to create new enums for the cooking condition, the cook type, etc. Even for things we didn't plan, like the student discount on Fridays, we had to update our class diagram to illustrate these key changes. We can also see these changes in our design patterns. At first, our design patterns were a bit basic since we had not yet implemented them. Now, the diagram accurately represents how these design patterns are used in our system and how they function.

## Third-Party Code vs. Original Code Statement:

We used some of Bruce Montgomery's Project 2 code as a base structure/template for this project. This includes the Namer, Main, and Simulation classes and the Utility and SysOut

interfaces. We altered these to match our system and goals. We also used some of Montgomery's code for simple functions in our DoughMasters class such as MoneyIn, MoneyOut, PayStaff, HireNewStaff, etc.

However, things like the customer, pizza, and inventory classes were implemented by us. Furthermore, the staff creation, cooking conditions, and pizza add-ons were also original and implemented using design patterns like factory, strategy, and decorator patterns. In summary, we used Montgomery's code as a base and then branched out with our own code the further we got into the project.

### **Statement of OOAD Process:**

A positive key design process element we experienced was developing our front-end prototype interface on Figma. We all had experience with Figma so there was no real negative learning curve that we experienced. Developing our interface went smoothly and was enjoyable, and is an excellent way to present a product to someone who did not work on the system.

One key design process issue we faced was implementing the inventory for our system. It was difficult to make pizzas each with different toppings and update the inventory based on the toppings chosen (since all pizza choices and toppings were randomized). To combat this problem we used Map in Java along with hashing. None of us had experience with these so some outside research was required.

Another positive key design process element we experienced was using design patterns. Although they were required, they helped immensely in organizing and making our code more readable. Using these design patterns led to us running into very few bugs, and the bugs we did run into were easily fixed since our system was loosely coupled.